Bilateral Filter

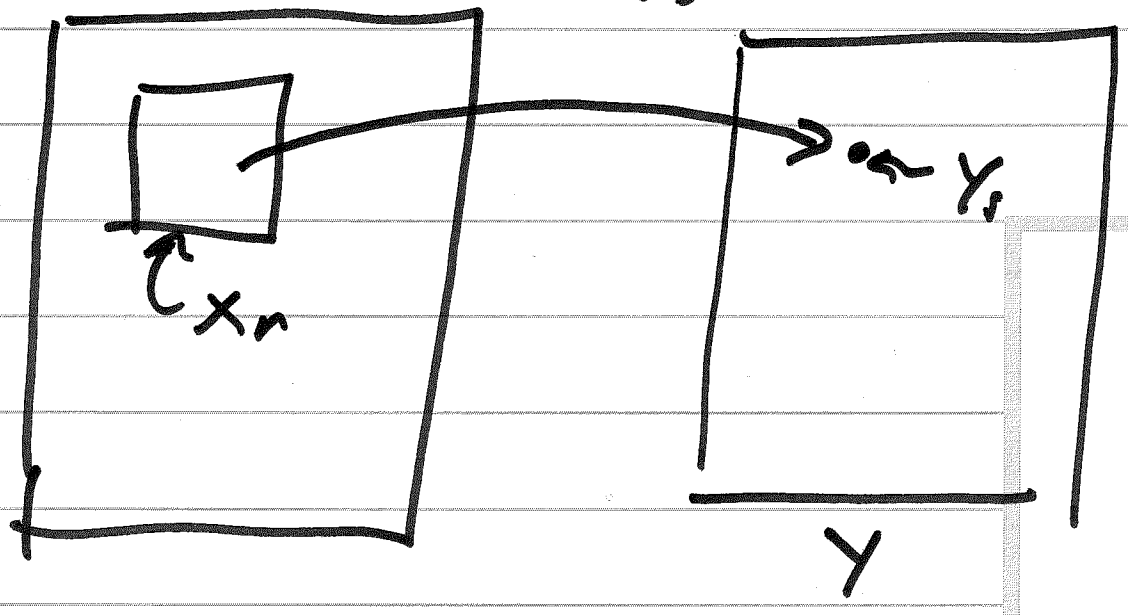
How to pick these

people use this in industry

$$Y_s = \sum_n w_{s,n} X_n$$ - Linear, Space-Varying

$w_{s,n}$ ↑ weight

$w_{s,n}$ should be large for pixels that are:
1) close in space
2) close in value



$$Y_s = \sum_r w_{s-r} X_r$$ - LSI

w_{s-r}: difference

bilateral filter

weighting that is dependent on spacing

spatial filter

gray levels

$$\tilde{w}_{s,n} = \exp\left\{-\frac{1}{2}\frac{\|s-n\|^2}{\sigma_x^2}\right\} \exp\left\{-\frac{1}{2}\frac{|x_s - x_n|^2}{\sigma_v^2}\right\}$$

$\tilde{w}_{s,n}$: Don't sum to 1.

σ_x^2: blur in space

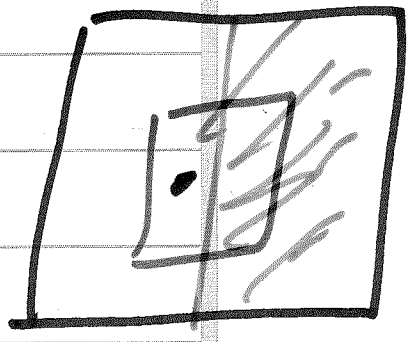
σ_v^2: blur in value — limited to pixels that are close

$$w_{s,n} = \frac{\tilde{w}_{s,n}}{\sum_{n'} \tilde{w}_{s,n'}}$$ - re-normalization

what is n'??

$$Y_s = \sum_n x_n w_{s,n}$$

EDGE Preserving



will not average over edges

$\vec{y} = W\vec{x}$

Time-invariant if $w_{i,j} = h_{i-j}$ for some h

$$W = \begin{bmatrix} \ddots & \ddots & \ddots \\ \ddots & 0 & b \\ \ddots & \ddots & \ddots \\ a & \ddots & \ddots \end{bmatrix}$$ – Toeplitz

$$y_s = \sum_r w_{s-r} x_r = \sum_r x_{s-r} w_r$$ convolution is communative

the most general form of an LSI filter (as long as it's closed)

FIR filter structure

$x^\infty = \lim_{n\to\infty} x_n$ – not "closed"

$x_n \longrightarrow [h_n] \longrightarrow y_n = h_n * x_n$

DTFT ↓

$x(e^{j\omega}) \longrightarrow [H(e^{j\omega})] \longrightarrow Y(e^{j\omega})$

complex sine waves - eigenvectors
$H(e^{j\omega})$ - eigenvalues

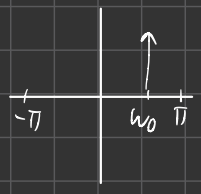
$$Y(e^{j\omega}) = X(e^{j\omega}) H(e^{j\omega})$$ amplitude & phase are adjusted

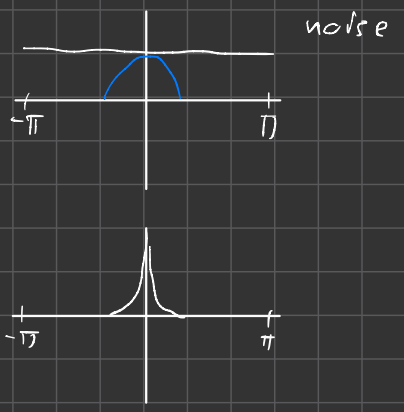
↑ DTFT

h_n - impulse responce

$x_n = e^{j\omega_0 n}$

$X(e^{j\omega}) = \delta(\omega - \omega_0), \quad |\omega| < \pi$

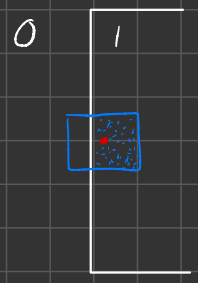


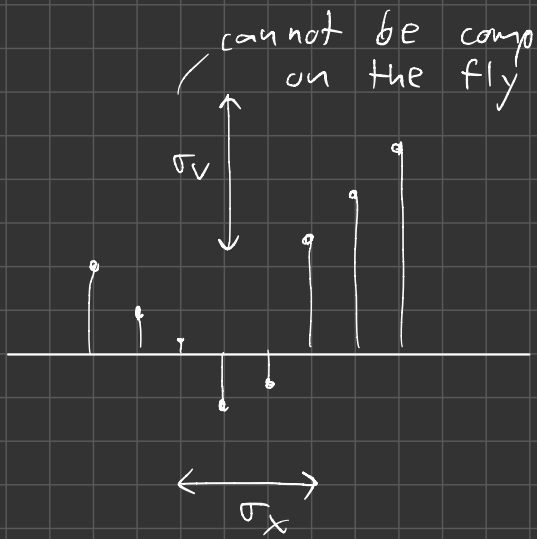


Non-linear filtering

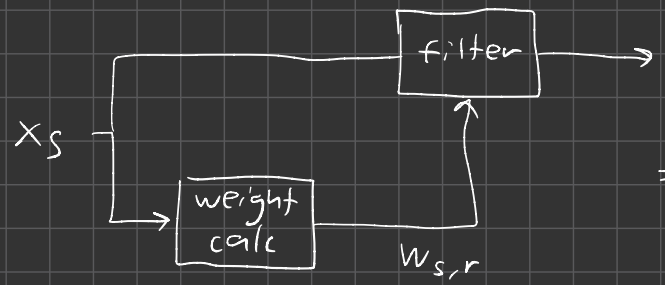
Idea: can $w_{s,r}$ be adjusted spatially

FT of Gaussian is also Gaussian-ish



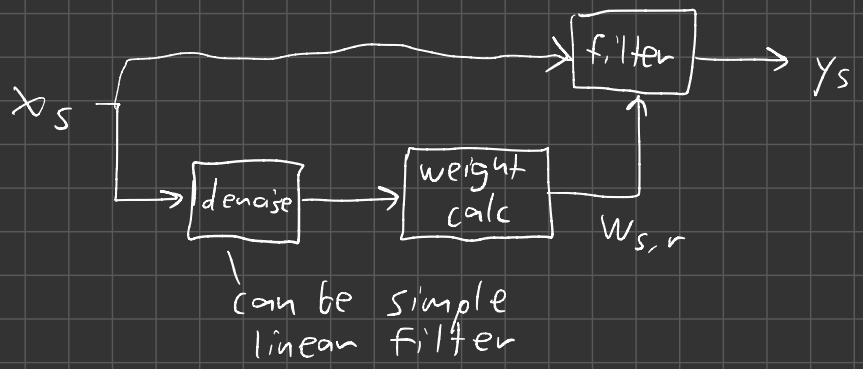


bilateral filter

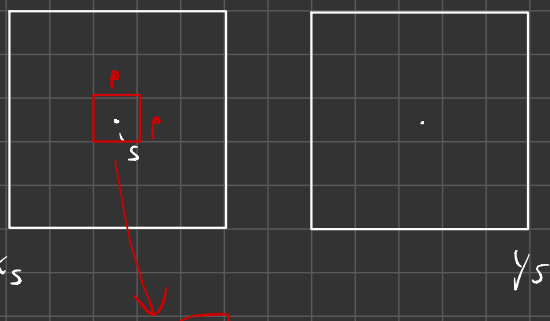


$$= \sum_r x_r w_{s,r}$$

variation (more practical)



Non-local means - not super practical (slow)



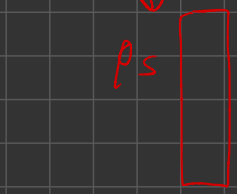
patches (vectors)

$$\tilde{w}_{s,r} = \exp\left\{-\frac{\|p_s - p_r\|^2}{2\sigma^2}\right\}$$

σ is sort of like representing the standard deviation of the noise you'd like to remove

$$\tilde{w}_{s,r} = \exp\left\{-\frac{1}{2\sigma^2\rho^2}\|p_s - p_r\|^2\right\}$$

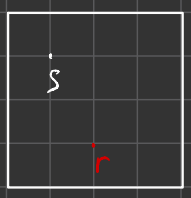
number of pixels in the patch



$p_s \in \mathbb{R}^{\rho^2}$ - window size

then:

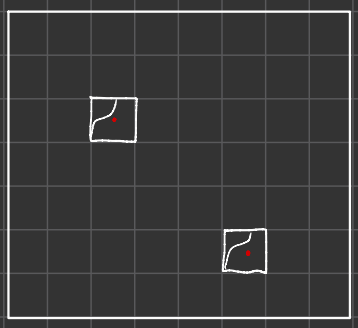
$$w_{s,r} = \frac{\tilde{w}_{s,r}}{\sum_{r'} \tilde{w}_{s,r'}}$$



$$\sum_r w_{s,r} x_r = y_s$$

$$\hookrightarrow y_s = \sum_r x_r w_{s,r}$$

p-hacking



local statistics tend to repeat (chair legs, tree branches, etc)

could also limit the search window size

Also,

BM3D
- software is online
- includes selecting patches, 3D FT's, etc

Non-local means } best
BM3D } un-trained

Deep Neural Net
- DNCNN